PU-learning disjunctive concepts in ILP

Hendrik Blockeel

KU Leuven, Department of Computer Science

Abstract.

PU learning is a variant of semi-supervised learning where labels of only one class are given. In this late-breaking paper, we present early results on an ILP method that PU-learns disjunctive concepts. The problem is motivated by some work on natural language learning, more specifically, learning the meaning of n-grams from (sentence, context)-pairs, where the context is described using first-order logic. An earlier solution for this task could learn only conjunctive concepts. We show experimentally that the novel method effectively learns disjunctive concepts from PU data. We also discuss challenges and limitations.

1 Introduction

Many systems for inductive logic programming learn classifiers in the form of rule sets, and more specifically, sets of definite Horn clauses. Such sets are typically learned in a supervised setting. An alternative learning setting is PU-learning, which stands for "learning from positive and unlabeled examples". PU-learners are given a set of instances E^+ that are known to be positive, and a set of instances $E^?$ for which it is not known whether they are positive or negative.

To our knowledge, PU-learning has not been considered in ILP. Learning from positives only has been considered [4], but is slightly different; it compares to PU-learning like supervised learning compares to semi-supervised learning.

In this short paper, we first briefly discuss PU-learning. We next show an ILP problem where PU-learning is relevant, and argue that it is beyond reach for standard PU-learners in two orthogonal ways: first, it is relational; second, a crucial assumption typically made by PU-learners is violated. We present an ILP algorithm that addresses both challenges and present some experimental results with it.

2 PU learning

PU-learners learn a binary classifier from two sets: a set of instances known to be positive, and a set of unlabeled instances, each of which may be positive or negative. Elkan and Noto [3] made the following observation, which has influenced recent work on PU-learning. Let $+$ denote the event that an instance is positive, and *pos* the event that an instance is *labeled* positive. Since only positives can be labeled positive, $P(pos|x) = P(pos|x, +)P(+|x)$.

Under the assumption that the labeled positives are a completely random subset of all positives (that is, each positive has the same probability k of being labeled), this reduces to $P(pos|x) = kP(+|x)$, and hence, $P(+|x) = P(pos|x)/k$. Thus, if k is known, it suffices to learn a probabilistic classifier that predicts whether x is labeled, which is a supervised learning task, and then dividing the predicted probabilities by k.

Most work on PU-learning implicitly or explicitly makes the "constant k" assumption. Under this assumption, PU-learning reduces to estimating the constant k and running a standard supervised learner. In the next section, we show a PU-learning problem where this assumption is clearly violated.

In: Nicolas Lachiche, Christel Vrain (eds.): Late Breaking Papers of ILP 2017, Orléans, France, September 4-6, 2017, published at http://ceur-ws.org

3 Relational grounded language learning

The motivating context for this work is the following task, "relational grounded language learning" (RGLL) [1, 2]. Given a set of *sentence/context* pairs, where *context* is a Datalog description of the context in which *sentence* was used, learn the circumstances under which an n-gram can be used (we call this the "meaning" of the n-gram). For instance, in one dataset [2], derived from Zitnick et al.'s "clip art" dataset [6], the sentence "a cat sits under the picnic table" accompanies the Datalog model

```
[object(o1),large(o1,c_table),color(o1,c_yellow),
object(o2),human(o2,c_boy),pose(o2,c_pose2),expression(o2,c_surprised),
object(o3),human(o3,c_girl),pose(o3,c_pose0),expression(o3,c_surprised),
object(o4),animal(o4,c_cat),size(o4,c_small),
object(o5),clothing(o5,c_hat),style(o5,c_pirate),act(o2, c_wear, o5),
object(o6),food(o6,c_pizza)]
```

which mentions many things, including a cat and a table. Creating for each model where "cat" occurs in the corresponding sentence, a clause of the form "cat ← *model*", and computing the least general generalization under theta-subsumption (briefly, "lgg") [5] of these clauses, might yield, e.g.,

$$\mathsf{cat} \leftarrow \mathsf{object(A),animal(A,c\_cat),size(A,c\_small)}$$

which summarizes the conditions under which the word "cat" can apparently be used (or: the "meaning" of "cat").

Earlier work [1] computed the meaning of an n-gram as the lgg of all the contexts where that n-gram was used, possibly allowing some exceptions [2] in order to handle noise. The disadvantage of that approach is that the lgg is a single clause, hence, a conjunctive concept. Such an approach cannot handle words with multiple meanings, which can be used in quite different contexts. For instance, in the dataset used, we expected the word "dog" to have only one meaning within the used dataset, just like "cat", but its occurrence in the bigram "hot dog" made it occur frequently in cases where the food, but not the animal, was present. As a result, the mentioned approaches could not learn the meaning of "dog"; they overgeneralized over dogs and hotdogs.

While ILP is perfectly capable of learning disjunctive concepts, RGLL faces the problem that a word does not *have* to be used when it is relevant. When the word "cat" occurs, a cat should be present (this is an assumption of RGLL), but when a cat is present, the sentence may not mention it. If we consider the occurrence of a word as a label for the context, it is clear that this is a PU-learning problem.

RGLL does not, however, fulfill the "constant k" condition. This condition would imply that the probability that a dog is mentioned, given that it appears, equals the probability that a hotdog is mentioned (using the spelling "hot dog", rather than "hotdog"), given that it appears. There is no reason to believe these are equal: some objects are more likely to be mentioned than others, there is the effect of alternative spellings or synonyms, etc. This is corroborated by the Zitnick dataset: e.g., the probability of having "dog" in the sentence, given that the context contains a dog or a hotdog, is 0.315 and 0.249, respectively.

For this reason, we developed a PU rule learner that does not assume a constant k. Instead, it assumes that the target concept is a disjunctive concept, and that the probability of being labeled is constant within each disjunct, but may differ from one disjunct to another.

4 PU-learning disjunctive concepts

4.1 PU-learning one rule

We first consider the case of PU-learning a conjunctive concept, which can be expressed using a single Horn clause. If there is no noise in the data, the clause must cover all positive examples. While it may cover unlabeled examples, the clause c that covers the fewest unlabeled examples is optimal: $P(pos|c)$ is maximal in this case, and therefore, so is $P(+|c) = P(pos|c)/k$.

The earlier approaches did not explicitly consider RGLL as a PU-learning task because, when learning conjunctive concepts from noiseless data, this is not necessary: the problem corresponds to computing an lgg. To allow for noisy data (sentences mentioning things not present in the context), [2] computed the lgg of "almost all" cases, rather than strictly all cases, by randomly generalizing the current clause with new positive instances until almost all positives are covered.

The PU-learning viewpoint gives a more principled method. Consider a sequence of clauses $c_1, c_2, \ldots, c_n$, where c_1 is a random positive instance and $c_i = lgg(c_{i-1}, e_i)$, with e_i a random positive instance not covered by c_{i-1}; we call this a *generalization path*.

Assume that the clauses $c_1, c_2, \ldots, c_j$ cover subsets of the target clause t, but c_{j+1} does not. Then $P(pos|c_i) = k$ for $i \leq j$, and $P(pos|c_i) < k$ for $i > j$, since labels occur entirely randomly within the disjunct, and there are no labeled examples outside the disjunct. Thus, even though we do not know k, we could in principle find k by constructing a curve that shows $P(pos|c_i)$ in terms of i, and looking what part of the curve is flat. The optimal c_i is the one just before the curve starts decreasing.

In practice, we cannot construct $P(pos|c_i)$; we can at best estimate it from a dataset as the proportion of instances covered by c_i that are labeled. This proportion randomly deviates from $P(pos|c_i)$. Furthermore, it tends to be biased positively because c_i is the result of repeated lggs of labeled instances. One way to compensate for the deviation is to construct a confidence interval for $P(pos|c_i)$ based on the proportion, and take the lower bound of this confidence interval; call this value $q(c_i)$. The confidence interval becomes narrower with increasing i, since the coverage of the clause increases with i. Assuming $P(pos|c_i)$ is constant for $i = 1, \ldots, j$, the expected value of $q(c_i)$ reaches a maximum for $i = j$. From that point of view, choosing the c_i that maximizes $q(c_i)$ is a good heuristic.

This leads to the following algorithm for PU-learning one rule (PULOR):

```
**procedure** PULOR:
    $c_1$ = random (not-yet-covered) labeled instance
    $i$=1
    stop=false
    **while** not stop:
        **repeat**
            $e$ = random labeled instance not covered by earlier rules
            $c_{i+1}$ = lgg($c_i$,$e$)
        **until** $c_{i+1} \neq c_i$ or repeat-loop was executed 5 times
        **if** $c_{i+1} = c_i$ **then** stop=true **else** $i$++
    **return** $\arg\max_{c_j, 1 \leq j \leq i} q(c_j)$
```

Our actual implementation of PULOR uses as heuristic not the q-function as defined above, but a monotonic function of it, namely the (lower bound of the) odds ratio of labeled versus unlabeled examples in the rule's coverage, as estimated using a sample of the unlabeled set. Figure 1 shows some typical curves. It confirms that using the lower bound of a confidence interval is advantageous; if the ratio itself is used, there is less generalization.

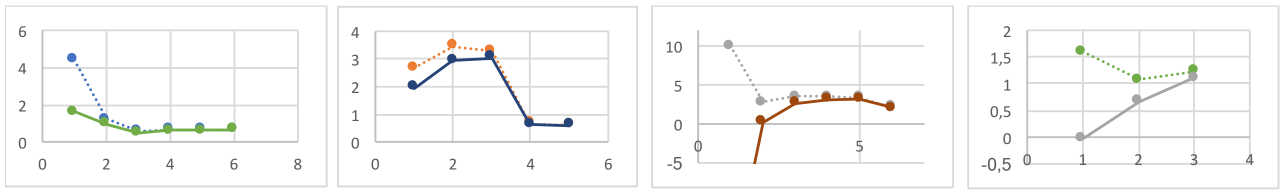

Fig. 1. Observed odds ratio (dotted) and the corresponding lower bound (solid), versus number of generalization steps, for some generalization paths

4.2 PU-learning a set of rules

Rule sets are often learned using a covering approach: learn one rule, mark the examples covered by that rule as covered, then repeat the process on the still uncovered examples. A single rule is typically learned by refining the rule until its accuracy is close to 1. In the PU-learning setting, the observed accuracy (the proportion of instances covered by the rule that have a positive label) differs from the real accuracy (the proportion of covered instances that are truly positive): if the true accuracy is a, the observed accuracy is ka.

If k were constant over all rules, one could try to determine it and then run a standard rule learner where the accuracy aimed for is k rather than 1. Since we assume different rules may have a different k value, k must be estimated per rule, but that can only be done once we know the rule. Here, therefore, the learning of the rule and the estimation of k must go hand in hand.

Assume the target is a disjunctive concept consisting of d disjuncts, where each disjunct can be accurately expressed as a single rule. Each such rule can be learned using PULOR.

Consider a generalization path that starts with a positively labeled example, which is part of disjunct i. Let us denote the "local" k value for the i'th disjunct as k_i. As long as the clauses cover a subset of that disjunct i, the expected proportion of labeled examples in the clause's coverage is k_i. As the clauses become increasingly general, they start covering negative instances and/or instances from other disjuncts. Clearly, covering negative instances decreases the observed accuracy, while covering instances from another disjunct j may increase or decrease it; this depends on whether $k_j > k_i$ or not.

Under these circumstances, it is not obvious that the q-maximization criterion that PULOR uses is optimal. When a generalization path starts in a disjunct with low k_i and at some point starts covering instances from a disjunct with high k_j without also covering too many negatives, the maximal q-value may be obtained for a clause that covers instances from multiple disjuncts. If rules are learned in the right order, from high to low k value, this will not occur.

Our algorithm for PU-learning a rule set, PULSE, uses the standard covering approach: it runs PULOR several times, marking covered examples as such before proceeding to learn the next rule. It does not attempt to optimize the order in which rules are learned (high to low k value).

5 Experiments

We have run PULSE on a dataset of 10000 sentence/context pairs, trying to learn rules for specific words or n-grams. Table 1 presents a sample from the outcomes. Whether a result is correct is somewhat subjective, there is often no agreed-upon ground truth; the best we can do is interpret the result. We cannot compare to other systems, because no other systems solve the considered task; for conjunctive concepts, however, PULSE often finds similar results as the earlier proposed ReGLL [2].

Note that PULSE sometimes finds sets of clauses where one clause is a specialization of another. This is a consequence of the fact that one can never be sure what the maximal reachable accuracy is (because k is unknown).

n-gram	**meaning**	**associated with**
tree	2/3: ob(A),lg(A,tree),spec(A,B),col(A,green),sz(A,big)	tree
apple tree	3/3: ob(A),lg(A,tree),spec(A,apples),col(A,green),sz(A,big)	apple tree
table	2/3: ob(A),lg(A,table),col(A,yellow)	table
	1/3: ob(A),lg(A,table),col(A,yellow) ;	
	ob(A),food(A,B),ob(C),lg(C,table),col(C,yellow)	table and food
dog	1/3: ob(A),animal(A,dog),col(A,brown),sz(A,small);	dog
	ob(A),lg(A,table),col(A,yellow),ob(B),food(B,hot_dog);	hotdog
	(3rd clause is a specialization of second)	
hot dog	1/3: ob(A),col(A,B),ob(C),food(C,hot_dog)	hotdog

Table 1. Some "meanings" learned using the proposed algorithm (predicate names and constants abbreviated for conciseness: ob=object, col=color, lg=large object, spec=species, sz=size, hu=human, ex=expression)

6 Conclusions

We have proposed a general algorithm for PU-learning sets of Horn clauses. The algorithm can trivially be extended to general rule learning. Contrary to existing work, it does not rely on the assumption that all positives are equally likely to be labeled. Applied to a language learning dataset, the method achieves interesting results: it is the first to learn disjunctive concepts in this setting, and makes an earlier method for handling noise obsolete. The proposed algorithm is highly preliminary: possible improvements include learning the rules in the optimal order, and estimating a rule's label proportion in a more principled manner. A more thorough experimental study is also warranted.

Acknowledgements

Research conducted while H.B. visited Université Jean Monnet, Saint-Etienne. H.B. thanks Elisa Fromont and Leonor Becerra-Bonache for making this stay possible, motivating this work, and for many interesting discussions.

References

1. L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet. A first-order-logic based model for grounded language learning. In *Proc. of IDA-2015*, pages 49–60. LNCS 9385, Springer, 2015.
2. L. Becerra-Bonache, H. Blockeel, M. Galván, and F. Jacquenet. Relational grounded language learning. In *Proc. of ECAI 2016*, pages 1764–1765, 2016.
3. C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proc. of KDD-2008*, pages 213–220, 2008.
4. Stephen Muggleton. Learning from positive data. In *Selected Papers from the 6th International Workshop on Inductive Logic Programming*, ILP '96, pages 358–376. Springer-Verlag, 1997.
5. G. D. Plotkin. *Machine Intelligence 5*, chapter A note on inductive generalization, pages 153–163. Edinburgh University Press, 1970.
6. C. L. Zitnick and D. Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 3009–3016. IEEE, 2013.